Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint Nr. 349

A Variational Taxonomy for Surface Reconstruction from Oriented Points

Christopher Schroers, Simon Setzer and Joachim Weickert

Saarbrücken 2014

Fachrichtung 6.1 – Mathematik
Universität des Saarlandes

Preprint No. 349
submitted: June 20, 2014

A Variational Taxonomy for Surface Reconstruction from Oriented Points

Christopher Schroers
Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Campus E1.7, 66041 Saarbrücken, Germany
schroers@mia.uni-saarland.de

Simon Setzer
Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Campus E1.7, 66041 Saarbrücken, Germany
setzer@mia.uni-saarland.de

Joachim Weickert
Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Campus E1.7, 66041 Saarbrücken, Germany
weickert@mia.uni-saarland.de

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-Mail: preprint@math.uni-sb.de
WWW: http://www.math.uni-sb.de/

Abstract

The problem of reconstructing a watertight surface from a finite set of oriented points has received much attention over the last decades. In this paper, we propose a general higher order framework for surface reconstruction. It is based on the idea that position and normal defined by each oriented point can be used to construct an implicit local description of the unknown surface. On the one hand, this allows us to systematically explain and relate several popular methods, for example implicit moving least squares, smooth signed distance surface reconstruction as well as (screened) Poisson surface reconstruction. On the other hand, it allows to derive and discuss a number of new approaches for reconstructing either the signed distance or the indicator function of the sought object. All of these approaches are able to achieve competitive results but one of them turns out to be especially promising. To improve reconstructions in difficult real world scenarios where point clouds have been estimated from colour images, we introduce a hull constraint that encourages the surface to stay within a given region. Our framework is implemented on the GPU using a recent cyclic scheme called Fast Jacobi, which combines low implementational effort with high efficiency.

1 Introduction

The problem of reconstructing a watertight surface from a finite set of oriented points is still a hot topic in computer vision and graphics although it has received an enormous amount of attention for over 25 years. An oriented point of a surface contains information about the position and the surface normal. Oriented point clouds can be obtained in numerous ways. For instance with active methods such as laser, structured light and time-of-flight scanning. Due to viewpoint dependence, usually many scans have to be acquired and subsequently aligned to cover the whole surface. Each surface measurement can conveniently be equipped with the direction to the source or an even better approximation of surface orientation by considering neighbouring measurements of one scan. Also many methods exist to estimate normals from point clouds, see e.g. [4]. Oriented point clouds can also directly arise in passive methods. A prominent example for this is the patch-based multi-view stereo reconstruction algorithm of Furukawa and Ponce [10].

Reconstructing an accurate surface is a difficult task. In practice it often occurs that some parts of the surface cannot be captured. Furthermore, one has to deal with uneven sampling due to overlapping scans, and the samples contain noise caused by inaccuracies of the sensor. Misalignments of the individual scans further increase the difficulty. The reconstruction problem is even more cumbersome when using multi-view stereo reconstruction algorithms on data captured from consumer grade cameras in uncontrolled environments. Due to this fact and because nowadays point clouds easily contain many millions of points, it is essential to apply robust and efficient algorithms. It is a common practice to fit the oriented points using a level set of an implicit function. Such methods can produce approximating surfaces, which is preferable if noise and outliers are present. Furthermore, they have the inherent advantage that one does not have to parameterise the surfaces. Many different approaches exist and commonly the implicit function is either an approximation to the indicator function or the signed distance function of the underlying surface.

In this paper, we develop a general higher order variational framework for surface reconstruction. It is based on the idea that each oriented point allows us to construct a function that provides a good local description of an implicit representation of the unknown surface. This framework allows us to reach two goals. First, we can systematically understand and classify a number of existing methods. Second, it enables us to derive novel approaches to surface reconstruction that are fairly simple and offer state-of-the-art performance. We show with the recent reconstruction Benchmark of Berger et al. [2] that one of these approaches yields favourable results when compared to the most popular and widely used methods, namely (screened) Poisson surface reconstruction and smooth signed distance surface reconstruction. Furthermore, we introduce a hull constraint that encourages the surface to stay within a given region. This improves reconstructions in difficult real world scenarios where point clouds have been estimated from color images.

Our paper is organised as follows. Chapter 2 covers the most important related work and Chapter 3 introduces our general framework with higher order terms. Chapter 4 then classifies existing approaches within this framework and Chapter 5 describes our novel approaches as well as the hull constraint. We discuss our GPU implementation in Chapter 6 followed by experimental results in Chapter 7. Finally, we give a conclusion and outlook in Chapter 8.

2 Related Work

We will focus on methods that fit the input data using a level set of an implicit function. As mentioned, the implicit function is either an approximation to the indicator function or the signed distance function of the underlying surface in many cases. Therefore, we use this as a criterion to broadly categorise prior work. Approaches that aim at recovering the structure of missing data, such as [3], are related in a wider sense as well.

Indicator Function Approximation. Kazhdan et al. estimate the indicator function by first computing a vector field that approximates the smoothed surface normal field and then integrating it in the least squares sense [13]. In [14], Kazhdan and Hoppe add an explicit point-wise constraint on the function value at the input points. Manson et al. reconstruct the indicator function using a wavelet basis [18]. As each sample point only influences a small number of coefficients, the reconstruction is very fast. Lempitsky and Boykov find a compromise between the number of collected input points and the surface area [16]. They minimise the resulting energy over binary functions using graph-cuts.

Signed Distance Function Approximation. A popular approach for estimating the signed distance function from a set of oriented points is the implicit moving least squares (IMLS) algorithm proposed by Shen et al. in [21]. Kolluri analyses a variant of this algorithm that uses constant basis functions [15]. He is able to show that it yields geometrically and topologically correct reconstructions if certain sampling conditions are fulfilled. In the presence of sharp features, it can make sense to use robust variants, see [9, 20]. Calakli and Taubin estimate an approximation of the signed distance function using a smoothing thin plate spline with additional pointwise constraints on the normals [5]. To minimise the energy, they employ a hybrid finite element / finite difference discretisation on an octree structure. Walder et al. consider the same energy but aim at expressing the solution as weighted sum of kernel functions centered at the input points [22]. If a triangle representation is desired in the end, an isosurface can be extracted from the implicit function with algorithms such as Marching Cubes [17].
As there exists a variety of different methods, it would be beneficial to have a joint platform that allows to explicitly display similarities and differences. From a didactic point of view, this gives an opportunity to explain existing methods to people new in this field. Furthermore, it offers a systematic Ansatz for deriving novel methods.

3 A General Higher Order Framework

In this section we describe how point and normal constraints can be used in a general variational formulation with higher order terms in order to reconstruct smooth surfaces. The reconstruction is then implicitly given by the boundary of the zero level set of the minimiser. To describe our framework, we first begin with some basic definitions.

Basic Definitions. Let us assume that a set of oriented points

$$\Big\{(\boldsymbol{p}_i, \boldsymbol{n}_i) \in \mathbb{R}^3 \times \mathbb{R}^3 \,\Big|\, i = 1, \dots, N\Big\} \tag{1}$$

has been sampled from a smooth surface. Here $\boldsymbol{p}_i$ and $\boldsymbol{n}_i$ denote location and normal, respectively. Then the surface can locally be approximated by a sufficiently small linear patch given by

$$\Big\{\boldsymbol{x} \in B_\sigma(\boldsymbol{p}_i) \,\Big|\, \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle = 0\Big\}, \tag{2}$$

i.e. a subset of the tangent plane to the surface at $\boldsymbol{p_i}$. Here $B_\sigma(\boldsymbol{p}_i)$ denotes an open ball with a small radius $\sigma > 0$ centred around $\boldsymbol{p}_i$. Accordingly, the signed distance function can also be well approximated locally around $\boldsymbol{p}_i$ by

$$f_i(\boldsymbol{x}) = \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle. \tag{3}$$

This gives rise to a data fidelity term

$$D(u) = \sum_{i=1}^{N} \int_{B_\sigma(\boldsymbol{p}_i)} \Big(u(\boldsymbol{x}) - f_i(\boldsymbol{x})\Big)^2 \, d\boldsymbol{x} \tag{4}$$

that rewards a close fit to the given functions by penalising a locally weighted squared L_2-distance to each of the functions $f_i(\boldsymbol{x})$. In the above equation, the deviation at each location within $B_\sigma(\boldsymbol{p}_i)$ is penalised with equal weight. In order to generalise this, we rewrite Equation 4 as

$$D(u) = \sum_{i=1}^{N} \int_{\Omega} w_\sigma(\|\boldsymbol{x} - \boldsymbol{p}_i\|) \Big(u(\boldsymbol{x}) - f_i(\boldsymbol{x})\Big)^2 \, d\boldsymbol{x}, \tag{5}$$

where

$$w_\sigma(s) = \begin{cases} 1, & \text{if } |s| \le \sigma \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The domain $\Omega \subset \mathbb{R}^3$ is a region that contains all local approximations and $u : \Omega \to \mathbb{R}$. In this notation, it becomes apparent that one can conveniently use an arbitrary weighting function instead of a hard window. An often preferable choice is a smooth, decaying function such as a Gaussian:

$$w_\sigma(s) = \exp\left(-\left(\frac{s}{\sigma}\right)^2\right). \tag{7}$$

This allows for good reconstructions even if coarser approximations with fewer linear patches are used, because the values further away from $\boldsymbol{p}_i$, that are usually less reliable, are only taken into account with a very small weight.

Higher Order Energy. So far, we have penalised deviations in function values only. In a general setting, we would now like to allow penalising the difference in all derivatives up to order K directly. To this end we first define a data term for the k-th derivative:

$$D_k(u) = \sum_{i=1}^{N} \int_{\Omega} \frac{w_{i,k}}{d_k} \left\| \mathcal{D}^{(k)} \Big(u - f_i \Big) \right\|^2 d\boldsymbol{x}, \tag{8}$$

where we omit the dependence on $\boldsymbol{x}$ for better readability. We allow choosing different weighting functions $w_{i,k}$ for each order of derivative and sample. As we will see, we can create pointwise, localised or global constraints by different choices of the weighting functions. The term d_k accounts for a possible normalisation. It can either be a constant or a function that allows for a pointwise reweighting. The term $\mathcal{D}^{(k)}$ is a differential operator that results in a vector of all derivatives of order k when applied to a function. Combining these data terms with a suitable smoothness functional $S(u)$ yields the energy

$$E(u) = \sum_{k=0}^{K} \alpha_k \, D_k(u) \; + \; \alpha \, S(u). \tag{9}$$

The weights $\boldsymbol{\alpha} = (\alpha_0, \dots, \alpha_K)^\top$ specify how strong deviations in each derivative should be penalised. This broad perspective gives a systematic way of approaching the surface reconstruction problem.

4 Relating Existing Methods within the Higher Order Framework

In this chapter, we will consider the cases $K = 0, 1, 2$ with $\alpha = 0$. This allows to explain and relate several popular surface reconstruction approaches.

4.1 Implicit Moving Least Squares ($K = 0$)

A choice of $K = 0$ means that we only consider a single data term that penalises the deviation in function values. It has the analytic solution

$$u(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} w_{i,0}(\boldsymbol{x}) \; \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle}{\sum_{i=1}^{N} w_{i,0}(\boldsymbol{x})}, \tag{10}$$

where one can choose a weighting function $w_{i,0}(\boldsymbol{x}) = a_i \, \exp(-\|\boldsymbol{x}-\boldsymbol{p}_i\|^2/\sigma_i^2)$ with a varying standard deviation σ_i modified by a normalisation factor a_i

for each input point. This corresponds to the *implicit moving least squares* (IMLS) algorithm with constant basis functions, which was propsed by Shen et al. [21]. The *partition of unity* (PoU) approach of Ohtake et al. [19] is very close in spirit and can even be equivalent for the case of constant basis functions. The weighting functions $w_{i,0}$ are carefully adapted for each individual sample. The choice of σ_i and a_i generally depend on the sampling density, which relates to the size of features one may expect. Since there is no smoothness term, a suitable choice of σ_i and a_i has to allow for removing isolated clutter and closing gaps. However, often it is not possible to find parameters that fulfil this while preserving details. Therefore IMLS-based reconstructions can exhibit spurious artefacts. Examples for this have also been shown in [5, 14] when comparing to the approach of Ohtake et al. [19].

4.2 (Screened) Poisson Surface Reconstruction ($K = 1$)

Let us start by only considering the first order data term, i.e. choosing $\boldsymbol{\alpha} = (0, 1)^\top$. Please recall that $f_i(\boldsymbol{x}) = \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle$ and thus $\boldsymbol{\nabla} f_i(\boldsymbol{x}) = \boldsymbol{n}_i$. This results in the energy

$$E(u) = \sum_{i=1}^{N} \int_{\Omega} \frac{w_{i,1}}{d_1} \|\boldsymbol{\nabla} u - \boldsymbol{n}_i\|^2 \, d\boldsymbol{x}, \tag{11}$$

where $w_{i,1}$ describes a local weighting, for example via a Gaussian function. We have chosen a pointwise normalisation factor $d_1(\boldsymbol{x})$ that accounts for uneven sample placements:

$$d_1(\boldsymbol{x}) = \sum_{i=1}^{N} w_{i,1}(\boldsymbol{x}). \tag{12}$$

The above energy can be rewritten as

$$E(u) = \int_{\Omega} \|\boldsymbol{\nabla} u - \boldsymbol{v}\,\|^2 \, d\boldsymbol{x} + R, \tag{13}$$

where the last expression R is a constant that contains residual terms that do not influence the minimiser, and $\boldsymbol{v}$ is a convex combination of the given normals:

$$\boldsymbol{v} = \sum_{i=1}^{N} \frac{w_{i,1}}{d_1} \boldsymbol{n}_i. \tag{14}$$

Interestingly, one can interpret this as solving two subsequent minimisation problems. First, a vector field is computed based on the input normals by solving a moving least squares problem similar as in the previous section but this time for the normal vectors. Then one finds the unknown function u in a second step by solving the above energy.

Poisson Surface Reconstruction. Obviously the energy in Equation 13 leads to a Poisson equation. For a suitable choice of $\boldsymbol{v}$ it resembles *Poisson surface reconstruction* [13]. The weighting required for this is motivated by an interesting observation: The gradient of the smoothed indicator function and the smoothed surface normal field are equal. Thus, $\boldsymbol{v}$ can be understood as an approximation of the smoothed surface normal field that can be obtained by performing a numerical integration

$$\boldsymbol{v} = \sum_{i=1}^{N} |\mathcal{P}_i| w_{i,1} \, \boldsymbol{n_i}, \tag{15}$$

where the patch sizes $|\mathcal{P}_i|$ are estimated with a density estimator. High densities relate to small patches and vice versa. While this seems to be a subtle change from Equation 14 to Equation 15, the effect is rather large: With this weighting, the length of the vectors $\boldsymbol{v}$ actually decreases towards zero when going away from the input points, as required for an indicator function. With the previous choice one estimates a distance field instead.
In both cases, the solution can only be computed up to a global constant. This generally does not have to pose a problem, but sometimes it is not possible to find a satisfactory one. Let us consider a somewhat artificial but very simple example to illustrate this. If the same surface orientation is measured everywhere, this will result in a constant vector field when using Equation 14. Thus, the reconstructed surface will be planar for any global offset. If the locations of the normals do not happen to be on a line, the reconstruction will thus appear to drift away from the input points. In practice, such problems have also been observed for Poisson surface reconstruction, where the input points were not fitted tightly enough and the reconstructions were too smooth.

Screened Poisson Surface Reconstruction. To account for this, Kazhdan and Hoppe add point constraints and show that this results in more accurate reconstructions [14]. With an appropriate α_0, they minimise the energy

$$E(u) = \alpha_0 \sum_{i=1}^{n} u(\boldsymbol{p}_i)^2 + \int_\Omega \|\boldsymbol{\nabla} u - \boldsymbol{v}\|^2 \, d\boldsymbol{x}, \tag{16}$$

over a suitable space of functions. The first term (screening term) of order zero can be obtained in our framework by choosing the Dirac distribution for each $w_{i,0}$. This effectively creates point constraints instead of constraints that are localised to some neighbourhood.

Relation to Global Optimisation for Shape Fitting. Another interesting observation is that the *global optimisation for shape fitting* approach of Lempitsky and Boykov [16] can be closely related to Poisson surface reconstruction. To see this, we rewrite Equation 13: We expand the scalar product, use the divergence theorem to perform partial integration, and introduce parameters α and β in the following way:

$$E(u) = \alpha \int_{\Omega} \|\boldsymbol{\nabla} u\|^{\beta+1} + 2u \cdot div\, \boldsymbol{v}\ d\boldsymbol{x} + \beta\ G(u). \tag{17}$$

The expression $G(u)$ is composed of three terms, where the first two are constant and do not have an effect in the minimisation. The last one influences the natural boundary condition. Explicitly $G(u)$ is given by

$$G(u) = R + \int_{\Omega} \|\boldsymbol{v}\|^2 d\boldsymbol{x} + \int_{\partial\Omega} \langle u\, \boldsymbol{v}, \boldsymbol{n} \rangle d\boldsymbol{s}. \tag{18}$$

Note that we recover Equation 13 for $\alpha = \beta = 1$. Setting $\beta = 0$ yields the energy minimised by Lempitsky and Boykov over binary functions.

Although very differently motivated, the vector field $\boldsymbol{v}$ is in both cases computed as a weighted sum of the normals using Gaussian weighting functions. In both methods, the standard deviation is adapted based on sample spacing as it is also common for moving least squares based approaches.

Lempitsky and Boykov minimise over a space of binary functions using graph cuts. In the context of surface reconstruction, the minimisation over binary functions has the drawback that it creates aliasing problems which have to be taken care of when extracting the isosurface.

4.3 Smooth Signed Distance Surface Reconstruction ($K = 2$)

Let us choose $K = 2$ in (9) and recall that $f_i(\boldsymbol{x}) = \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle$, $\boldsymbol{\nabla} f_i(\boldsymbol{x}) = \boldsymbol{n}_i$ and $\boldsymbol{H} f_i(\boldsymbol{x}) = 0$ with $\boldsymbol{H}$ denoting the Hessian. Let us choose

$$d_k = \sum_{i=1}^{N} \int_{\Omega} w_{i,k}(\boldsymbol{x})\, d\boldsymbol{x}, \tag{19}$$

so that each data term is automatically scaled by the area of all N weight functions. We select the Dirac distribution for all $w_{i,0}$ and $w_{i,1}$ creating pointwise constraints for input locations and normals. Furthermore, we set each $w_{i,2} := \frac{1}{N}$ which effectively creates a global constraint. This corresponds to the energy by Calakli and Taubin:

$$\begin{aligned} E(u) \; &= \frac{\alpha_0}{N} \sum_{i=1}^{N} \|u(\boldsymbol{p}_i)\|^2 \\ &+ \frac{\alpha_1}{N} \sum_{i=1}^{N} \|\boldsymbol{\nabla} u(\boldsymbol{p}_i) - \boldsymbol{n}_i\|^2 \\ &+ \frac{\alpha_2}{|\Omega|} \int_{\Omega} \|\mathbf{H} u(\boldsymbol{x})\|_F^2 \, d\boldsymbol{x}, \end{aligned} \tag{20}$$

which is minimised over a suitable function space [5]. Here $\| \cdot \|_F$ is the Frobenius norm. The last term can be understood as a smoothness term that arised by merely selecting appropriate weighting functions and thus it is in harmony with the constraints of the other terms.

5 Novel Formulations Derived from the General Framework

In this section we describe three novel variational formulations for the integration of point and normal constraints derived from our general framework. We will refer to them as Hessian-IMLS, TV-IMLS and HOM-IMLS. Furthermore, we also explain how we incorporate a hull constraint.

5.1 Hessian-IMLS

We have discussed that for pure IMLS-based approaches it is often not possible to find a σ to recover details while removing measurement errors and isolated clutter. Furthermore, we have argued that a Hessian smoothness term is in harmony with the point and normal constraints and it naturally arises for a suitable choice of weighting functions when discussing the case of $K = 2$. In general, this regulariser is popular for geometrical problems [22], also due to its good filling in behaviour in unsampled regions. It corresponds to the thin plate energy of order 2, cf. [8]. Thus we propose to minimise an energy combining both terms:

$$E(u) = \sum_{i=1}^{N} \int_{\Omega} w_i \big(u - f_i\big)^2 \, d\boldsymbol{x} + \alpha \int_{\Omega} \|\boldsymbol{H} u\|_F^2 \, d\boldsymbol{x}. \tag{21}$$

with

$$w_i = w_\sigma(\|\boldsymbol{x} - \boldsymbol{p}_i\|), \tag{22}$$

using a small constant $\sigma > 0$. We advocate that our model comprised of only two terms is the simplest choice possible within the higher order framework that incorporates the benefits of IMLS and an appropriate regularisation. In the SSD approach, the Dirac distribution as weighting function effectively removes the normal constraint from the first term in (20) in contrast to our second order model (21). We only require one data term that has the benefit of allowing to smooth out smaller measurement errors by considering local information with a small $\sigma > 0$. Additionally this term is easier to discretise as will be seen in Chapter 6. We do not require a hybrid finite element / finite difference discretisation as in [5], and our data term only contributes to the diagonal entries of the discrete system matrix. Furthermore, the use of weighting functions offers a natural way of treating varying patch sizes. They can arise e.g.when computing oriented points with the patch based multiview-stereo approach of Furukawa and Ponce [10].

Incorporating a Hull Constraint. It is often possible to estimate a hull that should contain the object to be reconstructed. In real world scenarios where oriented points are recovered from images, a prominent example of such a hull is the visual hull. Thus, we propose to augment our energy with an additional term that allows to encourage the surface to stay within a specified hull. This is especially helpful to steer the surface reconstruction when larger parts of the object are not sufficiently covered with oriented points. Let $\mathcal{H}$ denote the set of all points within a specified hull. Then we know that for any point $\boldsymbol{x}$ outside $\mathcal{H}$, its distance value $u(\boldsymbol{x})$ should be larger or equal to the euclidean distance from the hull $d(\mathcal{H}, \boldsymbol{x})$. However, inside the hull all values should be allowed without further penalisation. Thus, we propose to add the following term to our model (21):

$$\mathrm{Hull}(u) = \beta \int_{\Omega \setminus \mathcal{H}} \max\{0, d(\mathcal{H}, \boldsymbol{x}) - u(\boldsymbol{x})\}^2 d\boldsymbol{x}. \tag{23}$$

This hull constraint can be regarded as an optional additional level of control. Its importance can be specified by choosing a suitable $\beta > 0$ and it can be switched off by setting $\beta = 0$ if a hull should not be used.

5.2 TV-IMLS and HOM-IMLS

Instead of approximating the signed distance function, several other approaches approximate the indicator function of the unknown surface. This

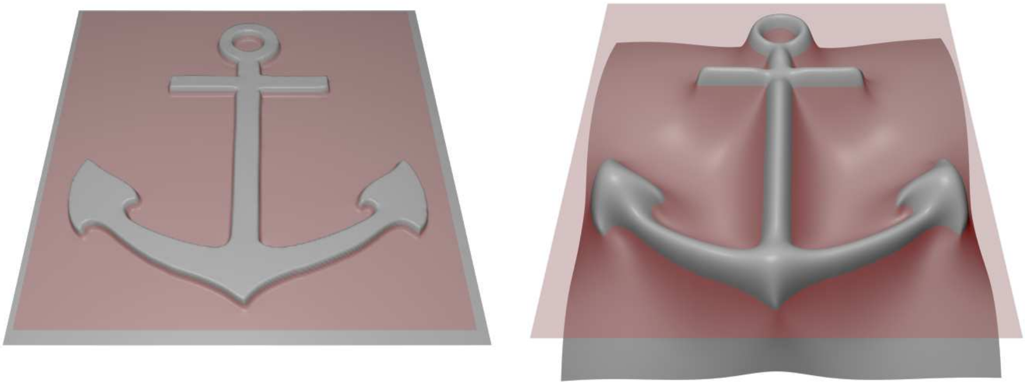

Figure 1: Reconstruction with our TV-IMLS **(left)** and Hessian-IMLS **(right)** approaches from two dimensional oriented points sampled from an anchor shape. The 2D implicit functions are visualised as height fields.

poses an interesing alternative. To achieve this, we reconsider the basic idea of understanding oriented points as local approximations within some small neighbourhood. As previously discussed, the oriented points allow us to construct a local approximation of the signed distance function

$$f_i(\boldsymbol{x}) = \langle \boldsymbol{x} - \boldsymbol{p}_i, \boldsymbol{n}_i \rangle. \tag{24}$$

However, this also allows to locally approximate the indicator function around $\boldsymbol{p}_i$ by $\Theta(f_i(\boldsymbol{x}))$, where Θ is a continuous approximation of the Heaviside function. Thus, we propose the data fidelity term

$$D_\Theta(u) = \sum_{i=1}^{N} \int_{B_\sigma(\boldsymbol{p}_i)} \Big(u(\boldsymbol{x}) - \Theta\big(f_i(\boldsymbol{x})\big) \Big)^2 \, d\boldsymbol{x}, \tag{25}$$

that rewards a close fit to the given local approximations. Also here, one can replace the hard window $B_\sigma(\boldsymbol{p}_i)$ by a Gaussian weighting function for example. A simple way to obtain a continuous approximation Θ is given by convolving the Heaviside function with a suitable kernel. If one uses a box function as convolution kernel, the resulting Θ can as well be interpreted as a scaled and truncated signed distance function as it is used in [7].

The indicator function is fundamentally different from the signed distance function as displayed in a simplified 2D example in Figure 1. This has to be considered in the choice of the smoothness term. As the gradient of the indicator function is zero almost everywhere, we propose to minimise a suitable energy with a first order smoothness term:

$$E_\Theta(u) = D_\Theta(u) + \alpha \int_\Omega \Psi(\|\boldsymbol{\nabla} u(\boldsymbol{x})\|^2)\; d\boldsymbol{x}, \tag{26}$$

where the parameter $\alpha > 0$ controls the degree of smoothness. This alternative is interesting as it only requires computing first order derivatives.
More specifically, we will consider two choices of Ψ, namely $\Psi(s^2) = s^2$ and $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ with a small constant $\epsilon > 0$. We refer to these models as HOM-IMLS and TV-IMLS, respectively. The former choice will lead to a penalisation of the squared gradient magnitude. The latter choice is motivated by the fact that it yields total variation regularisation (TV) for $\epsilon = 0$, which is well-suited for the reconstruction of piecewise constant functions such as an indicator function. Furthermore, it is known that TV penalises the perimeter of the level sets, which in this case corresponds to surface area. This is a favourable property, which usually leads to removal of small isolated clutter and reconstructions of low genus. This has been shown in the setting of range image integration by Zach et al. in [24]. When estimating an indicator function, it is straightforward to incorporate a hull constraint since the desired value outside the hull is known to be fixed.

6 GPU Implementation

Our GPU implementation uses the NVIDIA CUDA framework. It can essentially be divided into two stages of computation. First, setting up coefficients and right hand side for a system of equations and subsequently solving it. Let us now first discuss required discretisations before describing how both of these steps can efficiently be computed on parallel graphics hardware.
Either we set the domain Ω as a rectangular axis aligned bounding box $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \subset R^3$ that contains all oriented points or we manually specify it. The domain Ω is then discretised by choosing $(m_1, m_2, m_3)^\top$ equidistant samples in each direction resulting in $M = m_1 \cdot m_2 \cdot m_3$ unknowns.

Minimising Hessian-IMLS. Let us denote by $\boldsymbol{u}$ and $\boldsymbol{f}_i \in \mathbb{R}^M$ the discrete versions of $u(\boldsymbol{x})$ and $f_i(\boldsymbol{x})$. We have rearranged the unknowns in a vector using column major ordering. Let $\boldsymbol{W}_i$ be a diagonal matrix that contains all weights $w_i(\boldsymbol{x})$ and $V := \{x, y, z\}$. Then we can discretise Equation 21 as follows:

$$E(\boldsymbol{u}) = \sum_{i=1}^{N} \|\boldsymbol{W}_i^{\frac{1}{2}}(\boldsymbol{u} - \boldsymbol{f}_i)\|^2 + \alpha \sum_{j=1}^{M} \sum_{\gamma \in V^2} (\boldsymbol{D}_\gamma \boldsymbol{u})_j^2\,, \tag{27}$$

where $\boldsymbol{D}_{\boldsymbol{\gamma}}$ realises the corresponding second order derivative at each location where it can be estimated using a given stencil. The necessary and sufficient condition for a minimiser can be obtained in this case by setting the gradient to zero:

$$\left(\sum_{i=1}^{N} \boldsymbol{W}_i + \alpha \sum_{\gamma \in V^2} \boldsymbol{D}_{\boldsymbol{\gamma}}^{\top} \boldsymbol{D}_{\boldsymbol{\gamma}}\right) \boldsymbol{u} = \sum_{i=1}^{N} \boldsymbol{W}_i \boldsymbol{f}_i. \tag{28}$$

In the presence of a hull constraint, we need to consider the gradient of 23 additionally. The j-th component of it can be written as

$$2\,\beta\, H(d_j)\, H(d_j - u_j)\, (u_j - d_j), \tag{29}$$

where H denotes the Heaviside function and $\boldsymbol{d}$ the discrete version of $d(\mathcal{H}, \boldsymbol{x})$. Although the Heaviside function itself is discontinuous, the above expression is continuous.

Minimising TV-IMLS and HOM-IMLS. Similar as in the previous paragraph, we discretise (26) as follows:

$$E(\boldsymbol{u}) = \sum_{i=1}^{N} \|\boldsymbol{W}_i^{\frac{1}{2}}(\boldsymbol{u} - \boldsymbol{\Theta}(\boldsymbol{f}_i))\|^2 + \alpha \sum_{j=1}^{M} \Psi\left(\sum_{\gamma \in V} (\boldsymbol{D}_\gamma \boldsymbol{u})_j^2\right). \tag{30}$$

Here $\Theta(f_i(\boldsymbol{x})) \in \mathbb{R}^M$ is the discrete version of $\Theta(f_i(\boldsymbol{x}))$, and $\boldsymbol{D}_\gamma$ realises forward differences in direction of γ for each location where forward differences can be evaluated. By computing the gradient, we obtain the necessary condition a minimiser u must fulfil:

$$\left(\sum_{i=1}^{N} \boldsymbol{W}_i + \alpha \sum_{\gamma \in V} \boldsymbol{D}_\gamma^{\top} \boldsymbol{\Phi}(\boldsymbol{u}) \boldsymbol{D}_\gamma\right) \boldsymbol{u} = \sum_{i=1}^{N} \boldsymbol{W}_i\; \boldsymbol{\Theta}(\boldsymbol{f}_i), \tag{31}$$

where $\boldsymbol{\Phi}(\boldsymbol{u})$ is a diagonal matrix with

$$(\boldsymbol{\Phi}(\boldsymbol{u}))_{jj} = \Psi'\left(\sum_{\gamma \in V} (\boldsymbol{D}_\gamma \boldsymbol{u})_j^2\right). \tag{32}$$

Since the energy is strictly convex, the minimiser is unique and the necessary condition is also sufficient. More compactly, we express the nonlinear system as

$$(\boldsymbol{P} + \alpha\, \boldsymbol{A}(\boldsymbol{u}))\; \boldsymbol{u} = \boldsymbol{q}, \tag{33}$$

with the abbreviations

$$\boldsymbol{P} = \sum_{i=1}^{N} \boldsymbol{W}_i, \quad \boldsymbol{A}(\boldsymbol{u}) = \sum_{\gamma \in V} \boldsymbol{D}_\gamma^\top \boldsymbol{\Phi}(\boldsymbol{u}) \boldsymbol{D}_\gamma, \quad \boldsymbol{q} = \sum_{i=1}^{N} \boldsymbol{W}_i \boldsymbol{\Theta}(\boldsymbol{f}_i). \tag{34}$$

The nonlinear system of equations can be solved by the fixed point iteration

$$\big(\boldsymbol{P} + \alpha\, \boldsymbol{A}(\boldsymbol{u}^{\boldsymbol{k}})\big)\, \boldsymbol{u}^{k+1} = \,\boldsymbol{q} \quad (k \geq 0), \tag{35}$$

i.e. by solving a sequence of linear systems. Compared to the smooth signed distance surface reconstruction [5], we only require one data term instead of two. Moreover, we do not require a hybrid finite element / finite difference discretisation. This is because we effectively enforce similarity to a small oriented patch instead of a pointwise function value and normal constraint. Therefore, the data term only contributes to the diagonal of the system matrix and not to any off-diagonal entries as in [5].
In the case of HOM-IMLS, the squared gradient magnitude is penalised and the resulting system of equations is linear.

Implementation Details. For setting up the system matrix and the right hand side, we carry out the summations required to compute $\boldsymbol{P}$ and $\boldsymbol{q}$ in parallel using atomic operations. Due to the fact that we use a small constant σ this operation is extremely fast. For solving the linear system of equations that either arise directly or within the fixed point iteration for solving a nonlinear system, we use the cascadic Fast Jacobi (FJ) solver of Grewenig et al. [11]. The FJ solver is essentially a modified Jacobi over-relaxation (JOR) method, where the relaxation parameter is not fixed but varied in a cyclic way. Due to this, FJ is much more efficient than JOR but still as simple to implement. In particular, it is perfectly suited for parallelisation as it merely requires knowing values from the last iteration to compute the new iterations result. We use 3D CUDA arrays bound to textures or surfaces, which is well-suited in this scenario. It allows for fast read and write operations required in each iteration and makes use of efficient 3D caching. In the linear case, $4 \cdot M^3$ variables have to be stored. This corresponds to a memory usage of 2 GiB when using a volumetric grid of 512^3 voxels and 32-bit floating point accuracy. For the nonlinear case, the nonlinearities $\boldsymbol{\Phi}(\boldsymbol{u}^k)$ are stored as well. The reconstruction with 400^3 voxels shown in Figure 2 took 7.4 seconds on a GeForce GTX690. This illustrates the good performance of FJ on modern GPUs.

7 Experiments

For the examples taken from the Stanford scanning repository [1], raw data in range grid format is available. We estimate normals using the neighbouring pixels within each range scan. We have used cubic voxels and Gaussian weighting functions with a constant σ equal to the voxel size for each input point in all experiments.

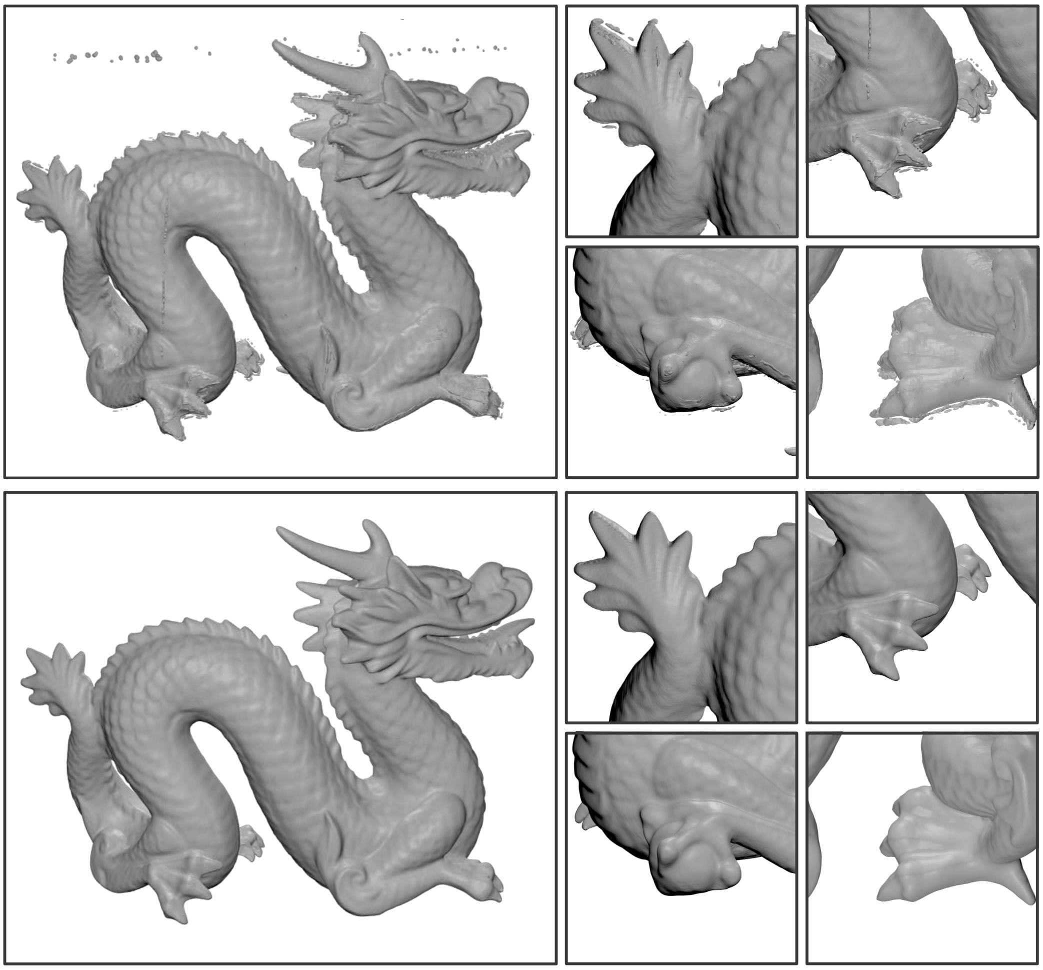

Figure 2: Reconstructions with our HOM-IMLS approach (26) at 400^3 voxels. **(top)** Choosing a smoothness weight close to zero ($\alpha = 0.1$) approximates a moving least squares solution, which results in isolated clutter. **(bottom)** By choosing a suitable smoothness weight ($\alpha = 10$), the isolated clutter is removed while small details are kept.

Dragon. In Figure 2, we have reconstructed the dragon model from the Stanford scanning repository [1] using our HOM-IMLS approach. One can see that choosing a too small smoothness weight ($\alpha = 0.1$) results in isolated clutter. This is reasonable because a smoothness weight of zero simply yields an implicit moving least squares solution which has been shown to produce reconstructions with spurious artefacts in many cases [14, 5]. By selecting a suitable smoothness weight ($\alpha = 10$), it is possible to remove the clutter while preserving the details even with the quadratic first order model. This illustrates the benefit of combining the strengths of the IMLS approach with the advantages of regularisation.

Drill. In Figure 3, we use use the drill dataset taken from the Stanford scanning repository [1] to compare wavelet surface reconstruction [18], (screened) Poisson surface reconstruction (PSR) [14], and smooth signed distance surface reconstruction (SSD) [5] to our novel approaches. We have always used the implementations provided by the respective authors and a tree depth of 9. Let us now consider the reconstructions in this order.
The wavelet approach **(b)** is very fast (0.4 seconds) and can also deal with noise and outliers to a certain extent. However, in this case it is not able to produce a faithful reconstruction. PSR **(c)** delivers a reasonable reconstruction, but the drill bit itself is rather unsmooth. Adding a screening weight ($\alpha_0 = 1$) in this case leads to overfitting noise **(d)**. In the SSD reconstruction **(g)**, the drill bit is also noisy although we have already chosen a large smoothness weight ($\alpha_2 = 25$) and one can clearly see that the top part is oversmoothed. By choosing a smaller weight for the smoothness term, it is possible to obtain a good reconstruction of the top part at the cost of more noise on the drill bit itself. All of our novel reconstructions **(e)**, **(f)** and **(h)** are able to convey the shape of the drill bit. Moreover, in the magnifications one can even slightly recognise the windings carved in towards the bottom of the drill bit. The running times of our HOM-IMLS, TV-IMLS, and Hessian-IMLS approaches are 2.9, 3.2, and 3.6 seconds, respectively. PSR requires 12.8 in the standard and 14.3 in the screened version, whereas SSD finishes in 4.8 seconds.

Torus. In Figure 4, we examine the different models and implementations for a simple shape defined by only few oriented points. PSR computes the solution on an octree that adapts to the input points. Doing so it is possible to obtain a reasonable but coarse reconstruction: see Figure 4 **(a)**. When switching off the adaptivity of the octree using a maximal tree depth of 8, artifacts occur around the input points, cf. Figure 4 **(b)**. Similar but

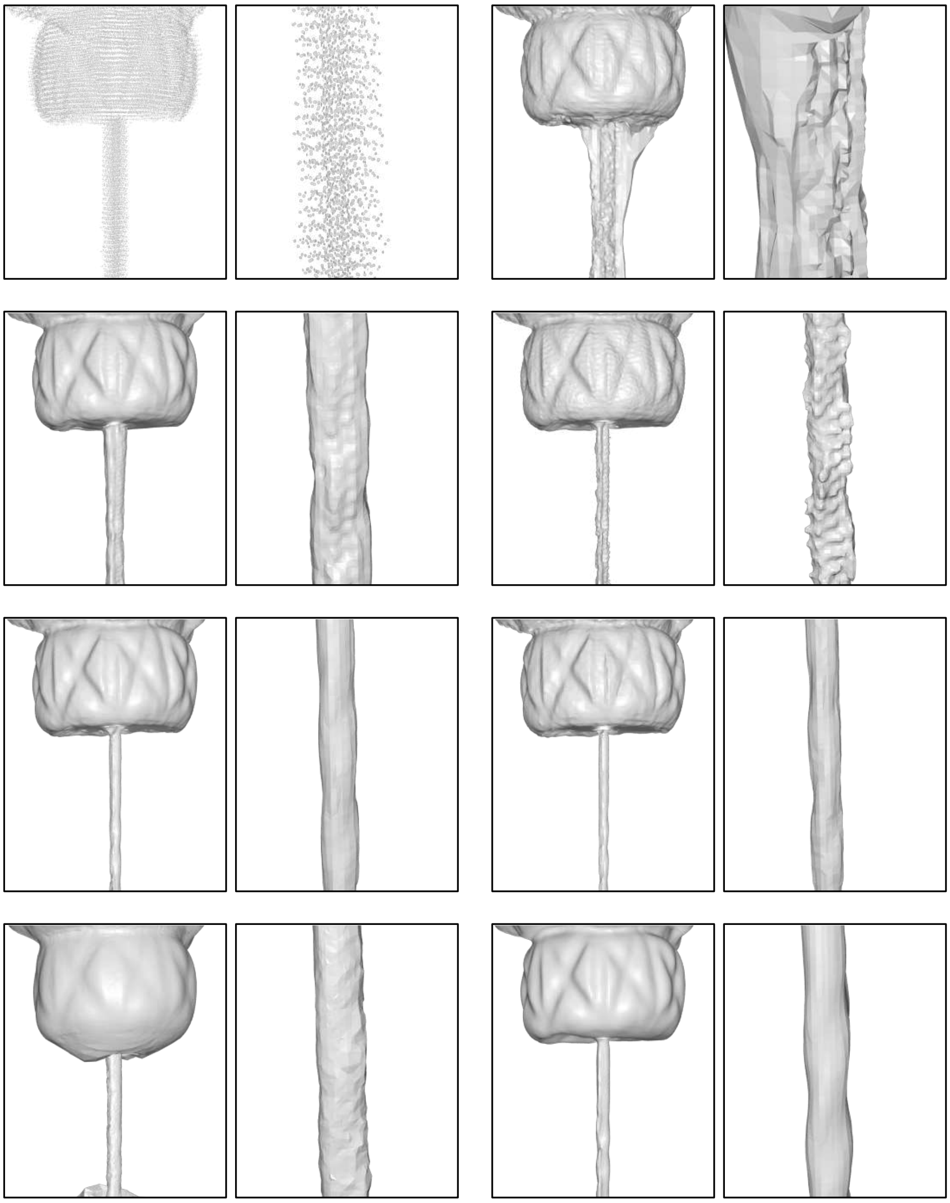

Figure 3: **In reading order**: **(a)** Oriented points **(b)** Wavelet surface reconstruction **(c)** Poisson surface reconstruction **(d)** Screened Poisson surface reconstruction **(e)** Our HOM-IMLS approach (26) **(f)** Our TV-IMLS approach (26) **(g)** SSD **(h)** Our Hessian-IMLS approach (21)

less prominent artifacts occur in our HOM-IMLS approach **(c)**. The SSD approach does not allow to switch off the adaptivity of the octree in the given implementation. Therefore, it can only compute a coarse reconstruction **(d)**.

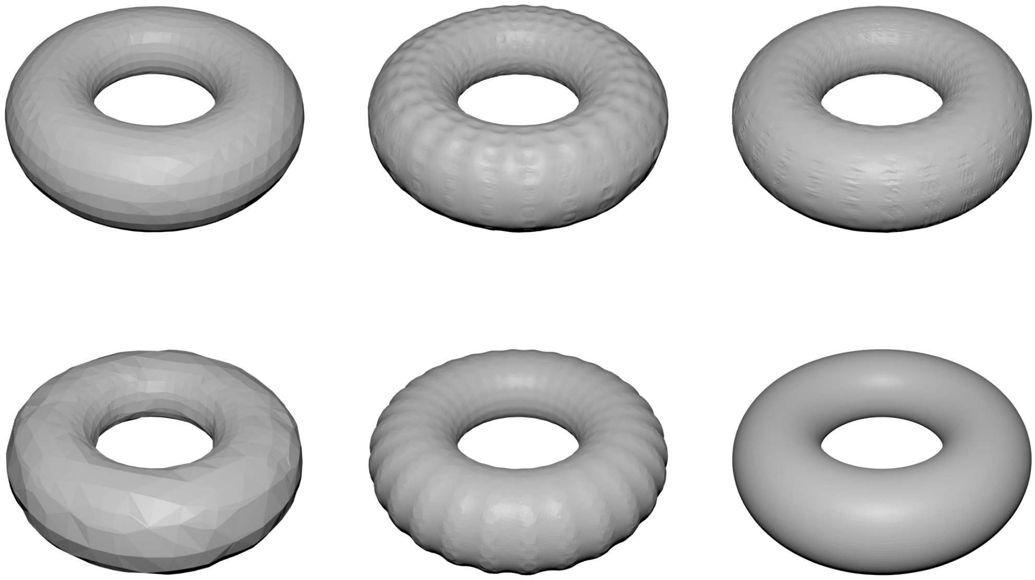

Figure 4: **In reading order**:**(a)** Screened PSR with adaptive octree **(b)** Screened PSR on fine regular grid **(c)** Our HOM-IMLS approach **(d)** SSD **(e)** Our TV-IMLS approach (26) **(f)** Our Hessian-IMLS approach (21)

In our reconstruction with TV-IMLS **(e)**, one can see how the surface area is minimised. However, here this property is not beneficial. Our Hessian-IMLS approach **(f)** achieves a reconstruction without artefacts and is in our eyes the most promising approach. Thus, we will focus on this approach in the remainder of the experiments. The coarse reconstructions with SSD (0.3 s) and PSR (0.4 s) are quite fast. However, two minutes are required when switching off the adaptivity of the octree in PSR. Our methods HOM-IMLS, TV-IMLS, and Hessian-IMLS finish in 0.9, 1.1, and 1.4 seconds, respectively.

Reconstruction Benchmark. We use the reconstruction benchmark of Berger et al. [2] for evaluating reconstruction accuracy of our Hessian-IMLS approach compared to (screened) PSR [14] and SSD [5]. The benchmark simulates scanner error as nonuniform sampling, noise and misalignment and covers many virtual scans of five different implicit surfaces. We use the most recently available author implementations for PSR and SSD, which are Versions 5.5 and 3.0, respectively. For PSR, we select the settings recommended by the authors for this benchmark, i.e. we use a screening weight of 4 for screened PSR and the same implementation with a screening weight of 0 to

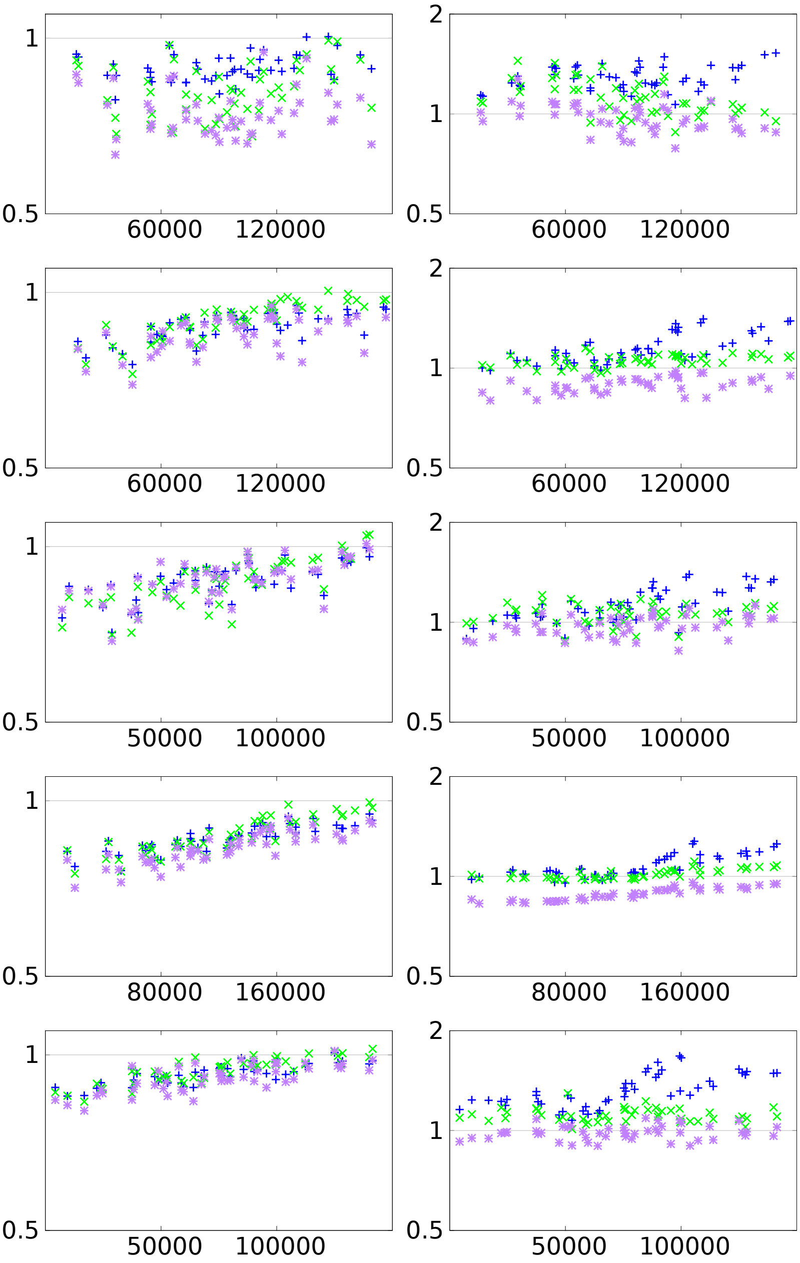

Figure 5: Reconstruction accuracy compared with the benchmark of Berger et al. [2]. For each of the five datasets, Anchor, Dancing, Daratech, Gargoyle and Quasimodo (rows from top to bottom), the two plots show the ratios of the mean distance (left) and mean normal errors (right) of screened PSR (blue), SSD (green) and our approach Hessian-IMLS (purple), relative to the original PSR algorithm. Each symbol corresponds to one benchmark test, where the horizontal axis denotes the amount of oriented points available in that test.

compute the unscreened version [14]. For SSD we have found the weights $\boldsymbol{\alpha} = (1, 1, 1)^\top$ to produce the best error values when considering both distance and angular errors. We use a resolution of 300^3 voxels, which is appropriate for covering present details in the models and accordingly choose an octree depth of 9 for the other approaches. For our approach we select a smoothness weight $\alpha = 1$.
When comparing the errors in distance values, one can see in Figure 5 that all methods tend to yield lower errors compared to PSR. In general, the error values are close together though. However, when considering the angular error values, our approach manages to also obtain the lowest error in most cases.

Globe. We have recorded several colour images of an ordinary globe and used VisualSFM [23] to estimate the camera poses. Subsequently we used the patch-based multiview stereo (PMVS) [10] algorithm to compute a coloured oriented point cloud. Figure 6 shows that our method is capable of producing a reasonable reconstruction from this point cloud despite of uncovered areas on the globe and noise. The texture has been computed by extending the idea of [6] to fit into our framework.

Hull Constraint. With the same pipeline as in the globe experiment, we have computed an oriented point cloud of a fountain: see Figure 7 **(a)**. In this case, the lack of oriented points in some locations, for example at the very top, causes unwanted filling in effects as in **(b)** and **(c)**. Our hull constraint according to Equation 23 allows for a better control of the surface **(d)**. In this case, we used a visual hull estimated from silhouettes made with the approach in [12].

Discussion. An octree as in [5, 14] allows for a better scaling in the unknowns and it is also possible to solve our model on an octree instead of a regular grid. However, in both previously mentioned implementations, the octree only adapts to the input data and not to the evolving solution, i.e. the unknown surface. Our fast GPU implementation allows to compute reconstructions of resolutions as required for the recent reconstruction Benchmark of Berger et al. [2] in a competitive runtime of a couple of seconds.

8 Conclusion and Outlook

We have proposed a general higher order framework for the implicit reconstruction of watertight surfaces from a finite set of oriented points and showed

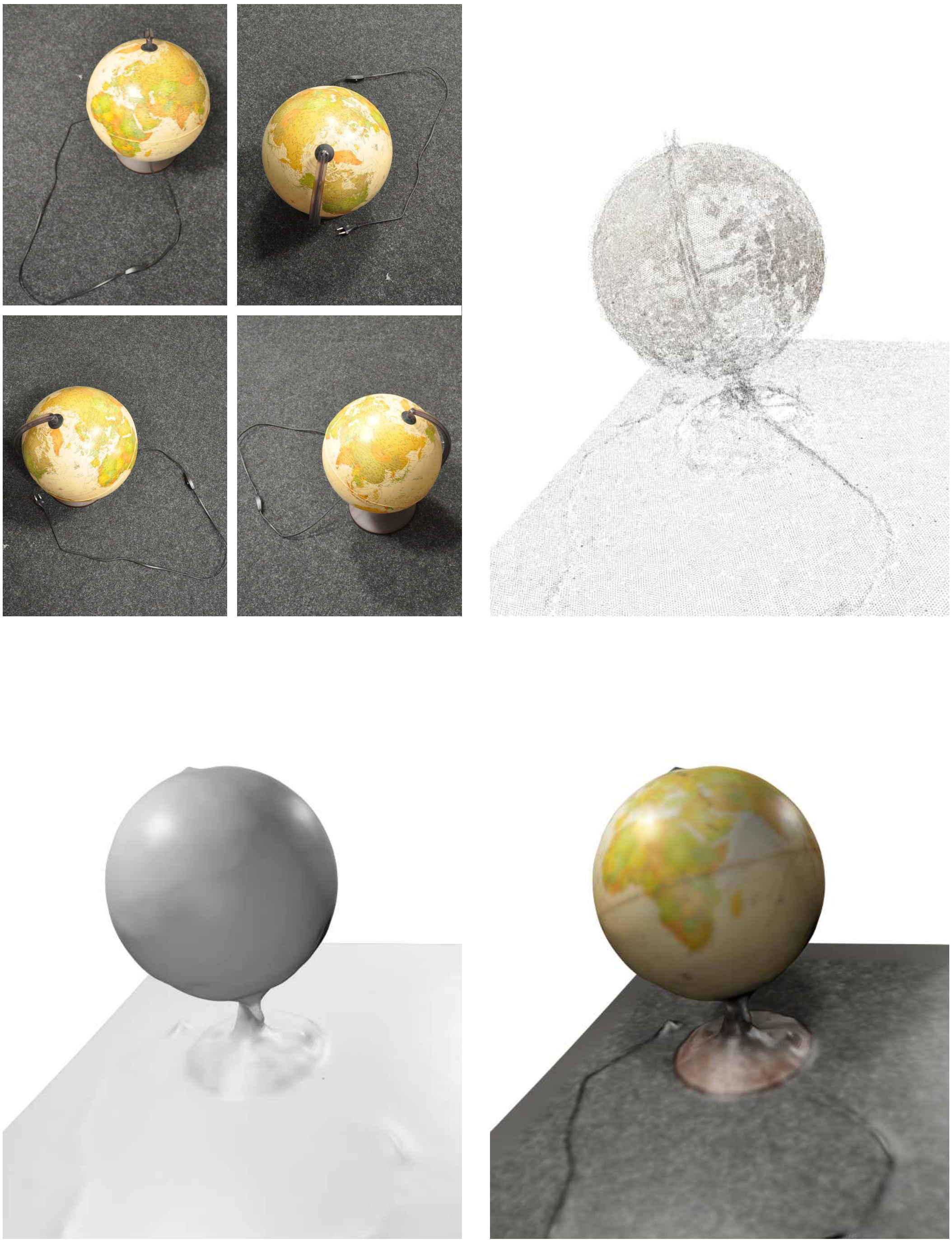

Figure 6: **In reading order**: **(a)** Subset of the input images **(b)** Coloured oriented point cloud produced by PMVS **(c)** Dense geometry reconstructed with our Hessian-IMLS approach (21) **(d)** Textured reconstruction

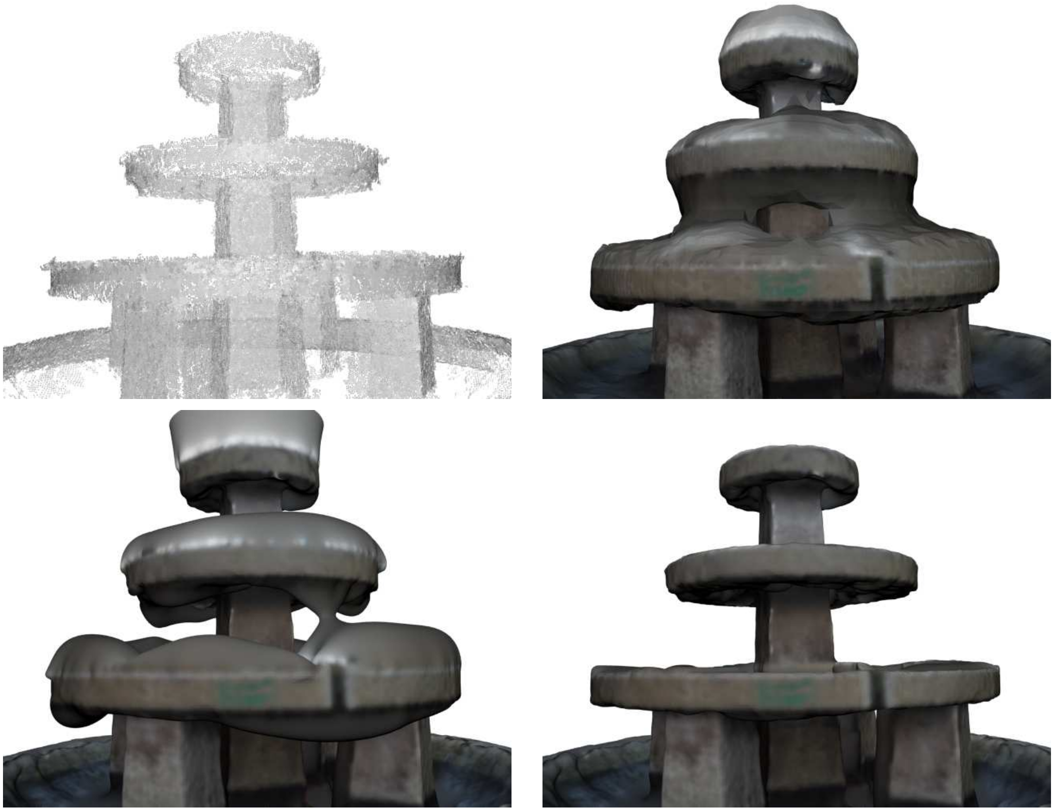

Figure 7: **In reading order**: **(a)** Input point cloud **(b)** SSD **(c)** Hessian-IMLS **(d)** Hessian-IMLS with hull constraint

the benefits of this systematisation: It makes specific features of popular existing approaches explicit. Moreover, it helps to identify gaps within the systematisation allowing to derive hitherto unexplored approaches. While all these approaches can yield competitive results, one of them showed to be especially promising. Our Hessian-IMLS formulation combines the benefits of implicit moving least squares based approaches and thin plate spline regularisation. In difficult real world scenarios, unwanted filling in effects that produce surfaces in regions that should be unoccupied can frequently appear for all approaches. To deal with such effects, we have proposed to incorporate a hull constraint.

We implemented our framework on the GPU using a novel cyclic scheme named Fast Jacobi for solving the resulting systems of equations. A possibility for future work would be a comprehensive study of different regularisers.

Acknowledgements. We would like to thank Stefanie Wuhrer for her advice, and we gratefully acknowledge funding by the Cluster of Excellence Multimodal Computing and Interaction.

References

[1] The Stanford 3D Scanning Repository. `http://graphics.stanford.edu/data/3Dscanrep/`.

[2] Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics*, 32(2):20:1–20:17, April 2013.

[3] Matthew Berger and Claudio T Silva. Medial Kernels. *Computer Graphics Forum*, 31(2pt4):795–804, 2012.

[4] A. Boulch and R. Marlet. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31(5):1765–1774, August 2012.

[5] F. Calakli and G. Taubin. SSD: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30(7):1993–2002, 2011.

[6] F. Calakli and G. Taubin. SSD-C: Smooth signed distance colored surface reconstruction. In John Dill, Rae Earnshaw, David Kasik, John Vince, and Pak Chung Wong, editors, *Expanding the Frontiers of Visual Analytics and Visualization*, pages 323–338. Springer London, 2012.

[7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH 96*, volume 3, pages 303–312, 1996.

[8] Jean Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In Walter Schempp and Karl Zeller, editors, *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100. Springer Berlin Heidelberg, 1977.

[9] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 544–552, New York, NY, USA, 2005. ACM.

[10] Y Furukawa and J Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[11] S. Grewenig, J. Weickert, C. Schroers, and A. Bruhn. Cyclic schemes for pde-based image analysis. Technical Report 327, Saarland University, Saarbrücken, Germany - Dept. of Math., March 2013.

[12] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[13] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70. Eurographics Association, 2006.

[14] M. Kazhdan and H. Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):29:1–29:13, July 2013.

[15] R. Kolluri. Provably good moving least squares. *ACM Trans. Algorithms*, 4(2):18:1–18:25, May 2008.

[16] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007.

[17] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[18] Josiah Manson, G. Petrova, and Scott Schaefer. Streaming surface reconstruction using wavelets. *Comput. Graph. Forum*, 27(5):1411–1420, 2008.

[19] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *ACM SIGGRAPH 2003 Papers*, pages 463–470, New York, NY, USA, 2003.

[20] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum*, 28(2):493–501, 2009.

[21] C. Shen, J. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers*, pages 896–904, New York, NY, USA, 2004.

[22] C. Walder, B. Schlkopf, and O. Chapelle. Implicit surface modelling with a globally regularised basis of compact support. *Computer Graphics Forum*, 25(3):635–644, 2006.

[23] Changchang Wu. VisualSFM : A Visual Structure from Motion System. `http://homes.cs.washington.edu/~ccwu/vsfm/`.

[24] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L^1 range image integration. In *Proc. Ninth International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society Press.